

Scilab Textbook Companion for  
Signals And Systems  
by S. Sharma <sup>1</sup>

Created by  
Shafquet Iftekhhar  
B.Tech (pursuing)  
Electronics Engineering  
Jamia Millia Islamia  
College Teacher  
Dr. Sajad A. Loan, JMI, New Delhi  
Cross-Checked by  
Giridharan, IITB

August 11, 2013

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

# Book Description

**Title:** Signals And Systems

**Author:** S. Sharma

**Publisher:** S. K. Kataria And Sons

**Edition:** 5

**Year:** 2006

**ISBN:** 81-85749-08-6

Scilab numbering policy used in this document and the relation to the above book.

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

**AP** Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

# Contents

List of Scilab Codes	4
1 Introduction to signals and systems	8
2 Linear Time Invariant System	25
3 Fourier Analysis of Periodic and APeriodic Continuous Time Signal	39
4 The Discrete Time Fourier Transform	44
5 Time and Frequency characterisation of signals and systems	48
6 Sampling And Laplace Transform	50
7 The Z Transform	55
8 Discrete Fourier Transform and Fast Fourier Transform	69

# List of Scilab Codes

Exa 1.1.a	Check for periodicity . . . . .	8
Exa 1.1.b	Check for periodicity . . . . .	8
Exa 1.4	Sketch and find power . . . . .	9
Exa 1.5	Sketch and find energy . . . . .	9
Exa 1.6	Sketch and find energy . . . . .	10
Exa 1.8	Find power of signal . . . . .	10
Exa 1.14.a	Check for causal system . . . . .	10
Exa 1.18.a	Check for time invariant systems . . . . .	11
Exa 1.18.b	Check for time invariant systems . . . . .	12
Exa 1.18.c	Check for time invariant systems . . . . .	12
Exa 1.19.a	Check for linear systems . . . . .	13
Exa 1.19.b	Check for linear systems . . . . .	14
Exa 1.20.b	Check for linear systems . . . . .	15
Exa 1.21	Check for linear systems . . . . .	16
Exa 1.22	Check for linear systems . . . . .	17
Exa 1.25	Check for linear systems . . . . .	18
Exa 1.27	Find energy of signal . . . . .	19
Exa 1.28	Find power of signal . . . . .	19
Exa 1.30	Find energy of signal . . . . .	19
Exa 1.31.a	Check for periodicity . . . . .	20
Exa 1.31.b	Check for periodicity . . . . .	20
Exa 1.33	Find power of signal . . . . .	20
Exa 1.34	Find energy of signal . . . . .	21
Exa 1.39.a	Sketch continous time signal . . . . .	21
Exa 1.39.b	Sketch continous time signal . . . . .	22
Exa 1.43	Check for linear systems . . . . .	22
Exa 1.47	Check for time invariant systems . . . . .	23
Exa 1.49.a	Check for periodicity . . . . .	24

Exa 1.49.b	Check for periodicity . . . . .	24
Exa 2.1	Convolution of two continuous time functions . . . . .	25
Exa 2.2	Find response of system . . . . .	26
Exa 2.3	Find unit step response of system . . . . .	26
Exa 2.4	Convolution of two continuous time functions . . . . .	27
Exa 2.5	Evaluation of output of LTI system . . . . .	28
Exa 2.6	Find response of system . . . . .	29
Exa 2.7	Convolution of two discrete time signals . . . . .	29
Exa 2.8	Find response of system . . . . .	30
Exa 2.17.a	Check for causal system . . . . .	31
Exa 2.17.b	Check for causal system . . . . .	31
Exa 2.17.c	Check for causal system . . . . .	31
Exa 2.19.a	Check for linear systems . . . . .	32
Exa 2.19.b	Check for linear systems . . . . .	33
Exa 2.21.a	Check for linear systems . . . . .	34
Exa 2.21.b	Check for linear systems . . . . .	35
Exa 2.25	Check for linear systems . . . . .	36
Exa 2.59	Convolution of two discrete time signals . . . . .	37
Exa 3.8	Fourier Transform . . . . .	39
Exa 3.9	Fourier Transform . . . . .	40
Exa 3.15	Fourier Transform . . . . .	41
Exa 3.31	Fourier Transform . . . . .	42
Exa 4.1	DTFT computation . . . . .	44
Exa 4.2	DTFT computation . . . . .	44
Exa 4.3	DTFT computation of unit impulse . . . . .	45
Exa 4.5	DTFT computation . . . . .	45
Exa 4.9	Sketch discrete time signal . . . . .	45
Exa 4.10	DTFT of cosine . . . . .	46
Exa 4.12	DTFT of unit step . . . . .	46
Exa 4.16	DTFT computation . . . . .	46
Exa 4.22	DTFT computation . . . . .	47
Exa 5.1	Bode Plot . . . . .	48
Exa 5.2	Bode Plot . . . . .	48
Exa 6.1	Find Nyquist rate . . . . .	50
Exa 6.2	Find Nyquist rate . . . . .	51
Exa 6.4	Find Nyquist rate . . . . .	51
Exa 6.26	Laplace transform of signal . . . . .	52
Exa 6.27.a	Laplace transform of function . . . . .	52

Exa 6.27.b Laplace transform of function . . . . .	53
Exa 6.27.c Laplace transform of function . . . . .	53
Exa 6.27.d Laplace transform of function . . . . .	53
Exa 6.27.e Laplace transform of function . . . . .	54
Exa 6.48 Find response of system . . . . .	54
Exa 7.1 z transform . . . . .	55
Exa 7.2 z transform of unit impulse . . . . .	55
Exa 7.3 z transform of unit step . . . . .	56
Exa 7.5 z transform of cosine . . . . .	56
Exa 7.6 z transform . . . . .	56
Exa 7.7 z transform of sequence . . . . .	56
Exa 7.10 z transform . . . . .	57
Exa 7.11.a z transform . . . . .	57
Exa 7.11.b z transform . . . . .	58
Exa 7.12 z transform using differentiation property . . . . .	58
Exa 7.13 z transform . . . . .	59
Exa 7.18 Find Discrete time input signal . . . . .	59
Exa 7.19 Inverse Z transform . . . . .	60
Exa 7.24.a Inverse Z transform using long division method . . . . .	60
Exa 7.36 z transform . . . . .	60
Exa 7.37 z transform . . . . .	61
Exa 7.38 z transform . . . . .	61
Exa 7.39 z transform using differentiation property . . . . .	61
Exa 7.42.a z transform of sequence . . . . .	62
Exa 7.42.b z transform of sequence . . . . .	62
Exa 7.43 z transform . . . . .	63
Exa 7.48.a z transform . . . . .	63
Exa 7.48.b z transform . . . . .	63
Exa 7.50 z transform of sequence . . . . .	64
Exa 7.52 z transform of discrete signal . . . . .	64
Exa 7.54 Inverse Z transform . . . . .	64
Exa 7.56 Find Discrete time input signal . . . . .	65
Exa 7.59.a z transform . . . . .	66
Exa 7.59.b z transform . . . . .	66
Exa 7.61.a z transform of discrete signal . . . . .	66
Exa 7.61.b z transform of discrete signal . . . . .	67
Exa 7.61.c z transform of discrete signal . . . . .	67
Exa 7.65 z transform . . . . .	67

Exa 7.68.a	z transform of discrete signal . . . . .	68
Exa 7.68.b	z transform of discrete signal . . . . .	68
Exa 8.1	Convolution of two finite duration sequences . . . . .	69
Exa 8.2	Responce of an FIR filter . . . . .	70
Exa 8.3	DFT and IDFT . . . . .	70
Exa 8.4	DFT computation . . . . .	70
Exa 8.5	DFT computation . . . . .	71
Exa 8.6	DFT of sequence . . . . .	71
Exa 8.7	DFT computation . . . . .	71
Exa 8.8	DFT computation . . . . .	72
Exa 8.9	IDFT computation . . . . .	72
Exa 8.10	IDFT computation . . . . .	72
Exa 8.11	DFT computation using FFT algorithm . . . . .	73
Exa 8.12	DFT computation using FFT algorithm . . . . .	73
Exa 8.13	DFT computation . . . . .	73
Exa 8.14	DFT computation . . . . .	74
Exa 8.15	DFT computation . . . . .	74
Exa 8.16	DFT computation . . . . .	74
Exa 8.17	DFT computation . . . . .	75



# Chapter 1

## Introduction to signals and systems

Scilab code Exa 1.1.a Check for periodicity

```
1 //Example 1.1a
2 //Determine whether the given signal is periodic or
  not
3 clc;
4 t=0:1/100:1
5 x=sin(15*%pi*t);
6 plot(x);
7 disp('ploting the signal and showing that it is
  periodic with period=2pi/15pi');
```

---

Scilab code Exa 1.1.b Check for periodicity

```
1 //Example 1.1b
2 //Determine whether the given signal is periodic or
  not
3 clc;
```

```

4 t=0:1/100:5
5 x=sin(sqrt(2)*%pi*t);
6 plot(x);
7 disp('ploting the signal and showing that it is
      periodic with period=2pi/sqrt(2)pi ');

```

---

#### Scilab code Exa 1.4 Sketch and find power

```

1 //Example 1.4
2 //Sketch the signal  $x(t)=A\sin(t)$ 
3 clc;
4 A=0.5;
5 t=0:1/100:10
6 x=A*sin(t);
7 plot(x);
8 //since it is a periodic signal so it is power
  signal
9 P=(integrate('((0.5)^2)*(sin(t)^2)', 't', 0, 2*pi))
  /(2*pi);

```

---

#### Scilab code Exa 1.5 Sketch and find energy

```

1 //Example 1.5
2 //Sketch the signal  $x(t)=A[u(t+a)-u(t-a)]$ 
3 clc;
4 A=1;
5 a=2;
6 t=-a:a
7 x=1;
8 plot(t,x)
9 //this signal is a finite duration signal so it is
  energy signal
10 E=integrate('1', 't', -a, a);

```

---

**Scilab code Exa 1.6** Sketch and find energy

```
1 //Example 1.6
2 //Sketch the signal  $x(t)=\exp(-a*t)$ 
3 clc;
4 t=0:1/100:10;
5 x=exp(-0.5*t);
6 plot(x)
7 E=integrate('(exp(-0.5*t)^2)', 't', 0, 10)
8 //Energy of the signal
```

---

**Scilab code Exa 1.8** Find power of signal

```
1 //Example 1.8
2 //Find the power of the signal  $x(t)=A\cos(Wo*t+\theta)$ 
3 clc;
4 A=20;
5 Wo=(2*%pi)/4;
6 for i=1:50
7     x(i)=A*cos(Wo*i);
8 end
9 p=0;
10 for i=1:4
11     p=p+(abs(x(i)^2))/4;
12 end
13 disp(p, 'The power of the given signal is =');
```

---

**Scilab code Exa 1.14.a** Check for causal system

```

1 //Example 1.14a
2 clc;
3 x=[1,2,3,4,0,4,3,2,1]
4 t=-length(x)/2:length(x)/2
5 count=0
6 mid=ceil(length(x)/2)
7 y=zeros(1,length(x))
8 y(mid+1:$)=x($:-1:mid+1)
9 for t=-1:-1:-mid
10     y(t+1+mid)=x(-t)
11 end
12 for i=1:length(x)
13     if(y(i)==x(i))
14         count=count+1
15     end
16 end
17 if(count==length(x))
18     disp('THE GIVEN SYSTEM IS CAUSAL')
19 else
20     disp('Since it depends on future values')
21     disp('THE GIVEN SYSTEM IS NON CAUSAL')
22 end

```

---

**Scilab code Exa 1.18.a** Check for time invariant systems

```

1 //Example 1.18a
2 clc;
3 t0=1;
4 T=10;
5 for t=1:T
6     x(t)=2*%pi*t/T;
7     y(t)=sin(x(t));
8 end
9 inputshift=sin(x(T-t0));
10 outputshift=y(T-t0);

```

```
11 if(inputshift==outputshift)
12     disp('THE GIVEN SYSTEM IS TIME INVARIANT')
13 else
14     disp('THE GIVEN SYSTEM IS TIME VARIANT');
15 end
```

---

**Scilab code Exa 1.18.b** Check for time invariant systems

```
1 //Example 1.18b
2 clc;
3 t0=2;
4 T=10;
5 for t=1:T
6     x(t)=t;
7     y(t)=t*x(t);
8 end
9 inputshift=x(T-t0);
10 outputshift=y(T-t0);
11 if(inputshift==outputshift)
12     disp('THE GIVEN SYSTEM IS TIME INVARIANT')
13 else
14     disp('THE GIVEN SYSTEM IS TIME VARIANT');
15 end
```

---

**Scilab code Exa 1.18.c** Check for time invariant systems

```
1 //Example 1.18c
2 clc;
3 t0=2;
4 T=10;
5 for t=1:T
6     x(t)=t;
7     y(t)=x(t)*cos(200*%pi*t);
```

```

8 end
9 inputshift=x(T-t0);
10 outputshift=y(T-t0);
11 if(inputshift==outputshift)
12     disp('THE GIVEN SYSTEM IS TIME INVARIANT')
13 else
14     disp('THE GIVEN SYSTEM IS TIME VARIANT');
15 end

```

---

#### Scilab code Exa 1.19.a Check for linear systems

```

1 //Example 1.19 a
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=t*x1(t)
12     y2(t)=t*x2(t)
13     y3(t)=t*x3(t)
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))

```

```

25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition
           principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

### Scilab code Exa 1.19.b Check for linear systems

```

1 //Example 1.19b
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=x1(t)^2
12     y2(t)=x2(t)^2
13     y3(t)=x3(t)^2
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');

```

```

26 disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition
           principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

### Scilab code Exa 1.20.b Check for linear systems

```

1 //Example 1.20b
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=x1(t)^2
12     y2(t)=x2(t)^2
13     y3(t)=x3(t)^2
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');

```



```

27 else
28     disp('It does not satisfy superposition
           principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

**Scilab code Exa 1.21** Check for linear systems

```

1 //Example 1.21
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a1=1;
6 b1=1;
7 a=7;
8 b=5;
9 for t=1:length(x1)
10     x3(t)=a1*x1(t)+b1*x2(t)
11 end
12 for t=1:length(x1)
13     y1(t)=a*x1(t)+b
14     y2(t)=a*x2(t)+b
15     y3(t)=a*x3(t)+b
16 end
17 for t=1:length(y1)
18     z(t)=a1*y1(t)+b1*y2(t)
19 end
20 count=0
21 for n=1:length(y1)
22     if(y3(t)==z(t))
23         count=count+1;
24     end
25 end
26 if(count==length(y3))
27     disp('It satisfy the superposition principle');

```

```

28 disp('THE GIVEN SYSTEM IS LINEAR ');
29 else
30     disp('It does not satisfy superposition
           principle ');
31     disp('THE GIVEN SYSTEM IS NON LINEAR');
32 end

```

---

**Scilab code Exa 1.22** Check for linear systems

```

1 //Example 1.22
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a1=1
6 b1=1
7 Wc=%pi
8 for t=1:length(x1)
9     x3(t)=a1*x1(t)+b1*x2(t)
10 end
11 for t=1:length(x1)
12     y1(t)=x1(t)*cos(Wc*t)
13     y2(t)=x2(t)*cos(Wc*t)
14     y3(t)=x3(t)*cos(Wc*t)
15 end
16 for t=1:length(y1)
17     z(t)=a1*y1(t)+b1*y2(t)
18 end
19 count=0
20 for n=1:length(y1)
21     if(y3(t)==z(t))
22         count=count+1;
23     end
24 end
25 if(count==length(y3))
26     disp('It satisfy the superposition principle');

```

```

27 disp('THE GIVEN SYSTEM IS LINEAR ');
28 else
29     disp('It does not satisfy superposition
        principle ');
30     disp('THE GIVEN SYSTEM IS NON LINEAR');
31 end

```

---

### Scilab code Exa 1.25 Check for linear systems

```

1 //Example 1.25
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a1=1;
6 b1=1;
7 a=7;
8 b=3;
9 for t=1:length(x1)
10     x3(t)=a1*x1(t)+b1*x2(t)
11 end
12 for t=1:length(x1)
13     y1(t)=a*x1(t)+b
14     y2(t)=a*x2(t)+b
15     y3(t)=a*x3(t)+b
16 end
17 for t=1:length(y1)
18     z(t)=a1*y1(t)+b1*y2(t)
19 end
20 count=0
21 for n=1:length(y1)
22     if(y3(t)==z(t))
23         count=count+1;
24     end
25 end
26 if(count==length(y3))

```

```
27 disp('It satisfy the superposition principle');
28 disp('THE GIVEN SYSTEM IS LINEAR ');
29 else
30     disp('It does not satisfy superposition
           principle ');
31     disp('THE GIVEN SYSTEM IS NON LINEAR');
32 end
```

---

**Scilab code Exa 1.27** Find energy of signal

```
1 //Example 1.27
2 //Energy of the signal  $x(t)=A\exp(-a*t).u(t)$ 
3 clc;
4 A=2;
5 a=0.5;
6 E=integrate(' (A*exp(-a*t))^2 ', 't', 0, 100); //Energy of
           the given signal
```

---

**Scilab code Exa 1.28** Find power of signal

```
1 //Example 1.28
2 //Power of the signal  $x(t)=A$ 
3 clc;
4 A=2;
5 P=(integrate('A^2 ', 't', 0, 100))/(2*100)
```

---

**Scilab code Exa 1.30** Find energy of signal

```
1 //Example 1.30
```

```
2 //Determine the energy of the signal  $x(n)=0.5^n$  for
   n >0
3 clc;
4 E=integrate('(0.5^n)', 'n', 0, 1000);
```

---

**Scilab code Exa 1.31.a** Check for periodicity

```
1 //Example 1.31a
2 //Determine whether the given signal is periodic or
   not
3 clc;
4 n=0:1/100:10
5 x=sin(6*%pi*n/7);
6 plot(x)//plotting the signal and showing it is
   periodic with period  $2\pi/(6\pi/7)$ ;
```

---

**Scilab code Exa 1.31.b** Check for periodicity

```
1 //Example 1.31b
2 //Determine whether the given signal is periodic or
   not
3 clc;
4 n=0:1/1000:100
5 x=sin(n/8);
6 plot(x);//plotting the signal and showing that it is
   periodic with period  $16\pi$ 
```

---

**Scilab code Exa 1.33** Find power of signal

```
1 //Example 1.33
```

```

2 //Find the power of the signal  $x(t)=A\cos(\omega t+\theta)$ 
3 clc;
4 A=10;
5 T=4;
6  $\omega = (2*\%pi)/T$ ;
7 for i=1:T
8     x(i)=A*cos( $\omega*i$ );
9 end
10 p=0;
11 for i=1:T
12     p=p+(abs(x(i)^2))/T;
13 end
14 disp(p, 'The power of the given signal is =');

```

---

**Scilab code Exa 1.34** Find energy of signal

```

1 //Example 1.34
2 //Find energy of  $x(t)=8\exp(2+i4\pi)t$ 
3 clc;
4 E=0;
5 for t=1:100
6     x(t)=8*exp((2+(%i*4*%pi))*t);
7 end
8 for t=1:100
9     E=E+x(t)^2;
10 end

```

---

**Scilab code Exa 1.39.a** Sketch continuous time signal

```

1 //Example 1.39 a
2 //Sketch the signal  $x(t)=u(t)$ 
3 clc;
4 t=0:1/100:10

```

```
5 x=1;
6 plot(t,x);
```

---

**Scilab code Exa 1.39.b** Sketch continuous time signal

```
1 //Example 1.39b
2 //Sketch the signal x(t)=tu(t)
3 clc;
4 t=0:1/100:10
5 x=t
6 plot(t,x)
```

---

**Scilab code Exa 1.43** Check for linear systems

```
1 //Example 1.43
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=x1(t)^2
12     y2(t)=x2(t)^2
13     y3(t)=x3(t)^2
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
```

```

20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition
           principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

**Scilab code Exa 1.47** Check for time invariant systems

```

1 //Example 1.47
2 clc;
3 k0=2;
4 n0=2;
5 N=10;
6 x=[1,2,3,4,5,6,7,8,9,10];
7 y=zeros(1,length(x));
8 for n=1:length(x)/k0
9     y(n)=x(k0*n);
10 end
11 inputshift=x(N-n0);
12 outputshift=y(N-n0);
13 if(inputshift==outputshift)
14     disp('THE GIVEN SYSTEM IS TIME INVARIANT')
15 else
16     disp('THE GIVEN SYSTEM IS TIME VARIANT');
17 end

```

---



**Scilab code Exa 1.49.a** Check for periodicity

```
1 //Example 1.49 a
2 //Determine whether the signal  $x(n)=\sin(7/9*\pi*(n^2)+1)$ 
   +1)
3 clc;
4 n=0:1/100:5
5 x=sin((7/9)*%pi*(n2)+1)
6 plot(x);
7 disp('this shows that signal is NOT periodic');
```

---

**Scilab code Exa 1.49.b** Check for periodicity

```
1 //Example 1.49 b
2 //Determine whether the signal  $x(n)=\cos(\pi*n/2)\cos(\pi*n/4)$ 
   pi*n/4)
3 clc;
4 n=0:1/100:100
5 x0=cos((%pi*n/2)+(%pi*n/4))
6 x1=cos((%pi*n/2)-(%pi*n/4))
7 x=(x0+x1)/2;
8 plot(x);
9 disp('plot shows that this is a periodic signal');
```

---

## Chapter 2

# Linear Time Invariant System

Scilab code Exa 2.1 Convolution of two continuous time functions

```
1 //Example 2.1
2 clc;
3 t=-8:1/100:8;
4 for i=1:length(t)
5     x(i)=exp(-t(i)^2);
6     h(i)=3*t(i)^2;
7 end
8 y=convol(x,h);
9 figure
10 plot2d(t,h);
11 title('Impulse response');
12 figure
13 plot2d(t,x);
14 title('Input signal');
15 figure
16 t2=-16:1/100:16
17 plot2d(t2,y);
18 title('Output signal');
```

---

**Scilab code Exa 2.2** Find response of system

```
1 //Example 2.2
2 clc;
3 t=-8:1/100:8;
4 for i=1:length(t)
5     if t(i)<0 then
6         x(i)=0;
7         h(i)=0;
8     else
9         x(i)=exp(-3.*t(i));
10        h(i)=1;
11    end
12 end
13 t1=t+1;
14 y=convol(x,h);
15 figure
16 plot2d(t1,h);
17 title('Impulse response');
18 figure
19 plot2d(t,x);
20 title('Input signal');
21 figure
22 t2=-16:1/100:16
23 plot2d(t2,y);
24 title('Output signal');
```

---

**Scilab code Exa 2.3** Find unit step response of system

```
1 //Example 2.3
2 clc;
3 R=100;
4 L=100;
5 t=-8:1/100:8;
6 for i=1:length(t)
```

```

7     if t(i)<0 then
8         x(i)=0;
9         h(i)=0;
10    else
11        h(i)=(R/L)*exp(-(R/L).*t(i));
12        x(i)=1;
13    end
14 end
15 y=convol(x,h);
16 figure
17 plot2d(t,h);
18 title('Impulse response');
19 figure
20 plot2d(t,x);
21 title('Input signal');
22 figure
23 t2=-16:1/100:16
24 plot2d(t2,y);
25 title('Output signal');

```

---

#### Scilab code Exa 2.4 Convolution of two continuous time functions

```

1 //Example 2.4
2 clc;
3 t=-8:1/100:8;
4 for i=1:length(t)
5     x(i)=3*cos(2.*t(i));
6     h(i)=exp(-abs(t(i)));
7 end
8 y=convol(x,h);
9 figure
10 plot2d(t,h);
11 title('Impulse response');
12 figure
13 plot2d(t,x);

```

```

14 title('Input signal');
15 figure
16 t2=-16:1/100:16
17 plot2d(t2,y);
18 title('Output signal');

```

---

### Scilab code Exa 2.5 Evaluation of output of LTI system

```

1 //Example 2.5
2 clc;
3 Max_Limit=10;
4 h=ones(1,Max_Limit);
5 N2=0:length(h)-1;
6 a=0.5;//constant a>0
7 for t=1:Max_Limit
8 x(t)=exp(-a*(t-1));
9 end
10 N1=0:length(x)-1;
11 y=convol(x,h)-1;
12 N=0:length(x)+length(h)-2;
13 figure
14 a=gca();
15 plot2d(N2,h)
16 xtitle('Impulse Response','t','h(t)');
17 a.thickness=2;
18 figure
19 a=gca();
20 plot2d(N1,x)
21 xtitle('Input Response','t','x(t)');
22 a.thickness=2;
23 figure
24 a=gca();
25 plot2d(N(1:Max_Limit),y(1:Max_Limit))
26 xtitle('Output Response','t','y(t)');
27 a.thickness=2;

```

---

**Scilab code Exa 2.6** Find response of system

```
1 //Example 2.6
2 clc;
3 t=-8:1/100:8;
4 for i=1:length(t)
5     if t(i)<0 then
6         x(i)=exp(2.*t(i));
7         h(i)=0;
8     else
9         x(i)=0;
10        h(i)=1;
11    end
12 end
13 t1=t+3;
14 y=convol(x,h);
15 figure
16 plot2d(t1,h);
17 title('Impulse response');
18 figure
19 plot2d(t,x);
20 title('Input signal');
21 figure
22 t2=-16:1/100:16
23 plot2d(t2,y);
24 title('Output signal');
```

---

**Scilab code Exa 2.7** Convolution of two discrete time signals

```
1 //Example 2.7
2 clc;
```

```

3 n=-8:1:8;
4 for i=1:length(n)
5     x(i)=exp(-n(i)^2);
6     h(i)=3.*n(i)^2;
7 end
8 y=convol(x,h);
9 figure
10 plot2d3(n,h);
11 title('Impulse response');
12 figure
13 plot2d3(n,x);
14 title('Input signal');
15 figure
16 n1=-16:1:16
17 plot2d3(n1,y);
18 title('Output signal');

```

---

**Scilab code Exa 2.8** Find response of system

```

1 //Example 2.8
2 clc;
3 n=-8:1:8;
4 for i=1:length(n)
5     if n(i)<0 then
6         x(i)=2^n(i);
7         h(i)=0;
8     else
9         x(i)=0;
10        h(i)=1;
11    end
12 end
13 y=convol(x,h);
14 figure
15 plot2d3(n,h);
16 title('Impulse response');

```

```
17 figure
18 plot2d3(n,x);
19 title('Input signal');
20 figure
21 n1=-16:1:16
22 plot2d3(n1,y);
23 title('Output signal');
```

---

**Scilab code Exa 2.17.a** Check for causal system

```
1 //Example 2.17a
2 clc;
3 disp(' y[n]=3x[n-2]+3x[n+2] ');
4 disp('THE GIVEN SYSTEM IS NON-CAUSAL');
5 disp('Since the value of output depends on future
      input');
```

---

**Scilab code Exa 2.17.b** Check for causal system

```
1 //Example 2.17b
2 clc;
3 disp(' y[n]=x[n-1]+a*x[n-2] ');
4 disp('THE GIVEN SYSTEM IS CAUSAL');
5 disp('Since the value of output doesnt depends on
      future input');
```

---

**Scilab code Exa 2.17.c** Check for causal system

```
1 //Example 2.17c
2 clc;
```



```

3 disp(' y[n]=x[-n] ');
4 disp('THE GIVEN SYSTEM IS NON-CAUSAL');
5 disp('Since the value of output depends on future
      input ');

```

---

**Scilab code Exa 2.19.a** Check for linear systems

```

1 //Example 2.19a
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 a1=0.5
8 b1=0.5
9 for n=1:length(x1)
10     x3(n)=a*x1(n)+b*x2(n)
11 end
12 for n=1:length(x1)
13     y1(n)=a1*n*x1(n)+b1
14     y2(n)=a1*n*x2(n)+b1
15     y3(n)=a1*n*x3(n)+b1
16 end
17 for n=1:length(y1)
18     z(n)=a*y1(n)+b*y2(n)
19 end
20 count=0
21 for n=1:length(y1)
22     if(y3(n)==z(n))
23         count=count+1;
24     end
25 end
26 if(count==length(y3))
27     disp('It satisfy the superposition principle');
28     disp('THE GIVEN SYSTEM IS LINEAR ');

```

```

29 else
30     disp('It does not satisfy superposition
           principle ');
31     disp('THE GIVEN SYSTEM IS NON LINEAR');
32 end

```

---

**Scilab code Exa 2.19.b** Check for linear systems

```

1 //Example 2.19b
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for n=1:length(x1)
8     x3(n)=a*x1(n)+b*x2(n)
9 end
10 for n=1:length(x1)
11     y1(n)=exp(x1(n))
12     y2(n)=exp(x2(n))
13     y3(n)=exp(x3(n))
14 end
15 for n=1:length(y1)
16     z(n)=a*y1(n)+b*y2(n)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(n)==z(n))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else

```

```

28     disp('It does not satisfy superposition
        principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

### Scilab code Exa 2.21.a Check for linear systems

```

1 //Example 2.21
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=5*sin(x1(t))
12     y2(t)=5*sin(x2(t))
13     y3(t)=5*sin(x3(t))
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition

```

```

        principle ');
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end

```

---

### Scilab code Exa 2.21.b Check for linear systems

```

1 //Example 2.21b
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for t=1:length(x1)
8     x3(t)=a*x1(t)+b*x2(t)
9 end
10 for t=1:length(x1)
11     y1(t)=7*x1(t)+5
12     y2(t)=7*x2(t)+5
13     y3(t)=7*x3(t)+5
14 end
15 for t=1:length(y1)
16     z(t)=a*y1(t)+b*y2(t)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(t)==z(t))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition
        principle ');

```

```
29     disp('THE GIVEN SYSTEM IS NON LINEAR');
30 end
```

---

### Scilab code Exa 2.25 Check for linear systems

```
1 //Example 2.25
2 clc;
3 x1=[1,1,1,1]
4 x2=[2,2,2,2]
5 a=1
6 b=1
7 for n=1:length(x1)
8     x3(n)=a*x1(n)+b*x2(n)
9 end
10 for n=1:length(x1)
11     y1(n)=x1(n)^2
12     y2(n)=x2(n)^2
13     y3(n)=x3(n)^2
14 end
15 for n=1:length(y1)
16     z(n)=a*y1(n)+b*y2(n)
17 end
18 count=0
19 for n=1:length(y1)
20     if(y3(n)==z(n))
21         count=count+1;
22     end
23 end
24 if(count==length(y3))
25     disp('It satisfy the superposition principle');
26     disp('THE GIVEN SYSTEM IS LINEAR ');
27 else
28     disp('It does not satisfy superposition
29         principle ');
30     disp('THE GIVEN SYSTEM IS NON LINEAR');
```

30 end

---

**Scilab code Exa 2.59** Convolution of two discrete time signals

```
1 //Example 2.59
2 clc;
3 n=-8:1:8;
4 for i=1:length(n)
5     if n(i)<0 then
6         x(i)=0;
7         h(i)=0;
8     else
9         x(i)=1;
10        h(i)=2^n(i);
11    end
12 end
13 y=convol(x,h);
14 figure
15 a=gca();
16 plot2d3(n,h);
17 a.x_location='origin';
18 a.y_location='origin';
19 title('Impulse response');
20 figure
21 a=gca();
22 plot2d3(n,x);
23 a.x_location='origin';
24 a.y_location='origin';
25 title('Input signal');
26 figure
27 a=gca();
28 n1=-16:1:16
29 plot2d3(n1,y);
30 a.x_location='origin';
31 a.y_location='origin';
```

```
32 title('Output signal');
```

---

## Chapter 3

# Fourier Analysis of Periodic and APeriodic Continuous Time Signal

Scilab code Exa 3.8 Fourier Transform

```
1 clc ;
2 close ;
3 // Analog S i g n a l
4 A =1; // Ampl i tude
5 Dt = 0.005;
6 t = 0: Dt :10;
7 xt = exp(-A*t);
8 Wmax = 2* %pi *1; // Analog Fr equency = 1Hz
9 K = 4;
10 k = 0:( K /1000) :K;
11 W = k* Wmax /K;
12 XW = xt* exp (- sqrt ( -1)*t'*W) * Dt;
13 XW_Mag = abs(XW);
14 W = [-mtlb_fliplr(W),W(2:1001)]; // Omega f rom
    Wmax to Wmax
15 XW_Mag = [mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
16 [ XW_Phase ,db] = phasemag (XW);
```



```

17 XW_Phase =[-mtlbfliplr(XW_Phase),XW_Phase(2:1001)];
18 //Plotting Continuous Time Signal
19 figure
20 a = gca ();
21 a.y_location = "origin";
22 plot (t,xt);
23 xlabel ( ' t in sec. ' );
24 ylabel ( ' x ( t ) ' )
25 title ( ' Continuous Time Signal ' )
26 figure
27 // P l o t t i n g Magni tude Re spons e o f CTS
28 subplot (2 ,1 ,1);
29 a = gca ();
30 a.y_location = "origin";
31 plot (W, XW_Mag );
32 xlabel ( ' Fr equency i n Radians /
           S e c o n d s > W' );
33 ylabel ( ' abs (X(jW) ) ' )
34 title ( 'Magni tude Re spons e (CTFT) ' )
35 // P l o t t i n g Phase Reponse o f CTS
36 subplot (2 ,1 ,2);
37 a = gca ();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot (W, XW_Phase *%pi /180) ;
41 xlabel ( ' Fr equency i n Radians /
           S e c o n d s > W' );
42 ylabel ( ' <X(jW) ' )
43 title ( ' Phase Re spons e (CTFT) i n Radians ' )

```

---

### Scilab code Exa 3.9 Fourier Transform

```

1 //Example 3.9
2 clc;
3 clear;

```

```

4 A=1;
5 Dt=0.005;
6 T1=4;
7 t=-T1/2:Dt:T1/2;
8 for i=1:length(t)
9 xt(i)=A;
10 end
11 Wmax=2*%pi*1;
12 K=4;
13 k=0:(K/1000):K;
14 W=k*Wmax/K;
15 xt=xt';
16 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
17 XW_Mag=real(XW);
18 W=[-mtlbfliplr(W),W(2:1001)];
19 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
20 subplot(2,1,1);
21 a=gca();
22 a.data_bounds=[-4,0;4,2];
23 a.y_location="origin";
24 plot(t,xt);
25 xlabel('t in sec. ');
26 title('Continuous Time Signal x(t)');
27 subplot(2,1,2);
28 a=gca();
29 a.y_location="origin";
30 plot(W,XW_Mag);
31 xlabel('Frequency in Radians/Seconds');
32 title('Continuous time Fourier Transform X(jW)');

```

---

### Scilab code Exa 3.15 Fourier Transform

```

1 //Example 3.15
2 clc;
3 clear;

```

```

4 T1=2;
5 T=4*T1;
6 Wo=2*%pi/T;
7 W=[-Wo,0,Wo];
8 ak=(2*%pi*Wo*T1/%pi)/sqrt(-1);
9 XW=[-ak,0,ak];
10 ak1=(2*%pi*Wo*T1/%pi);
11 XW1=[ak1,0,ak1];
12 figure
13 a=gca();
14 a.y_location="origin";
15 a.x_location="origin";
16 plot2d3('gnn',W,XW1,2);
17 poly1=a.children(1).children(1);
18 poly1.thickness=3;
19 xlabel('W');
20 title('CTFT of cos(Wot)');

```

---

### Scilab code Exa 3.31 Fourier Transform

```

1 //Example 3.31
2 clc ;
3 clear;
4 R=10^3;
5 C=10^-3;
6 A=1/(R*C);
7 Dt=0.005;
8 t=0:Dt:10;
9 xt=A*exp(-A*t);
10 Wmax=2*%pi*1;
11 K=4;
12 k=0:(K/1000):K;
13 W=k*Wmax/K;
14 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
15 XW_Mag=abs(XW);

```

```

16 W=[-mtlbfliplr(W),W(2:1001)];
17 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
18 [XW_Phase,db]=phasemag(XW);
19 XW_Phase=[-mtlbfliplr(XW_Phase),XW_Phase(2:1001)];
20 figure
21 a=gca();
22 a.y_location="origin";
23 plot(t,xt);
24 xlabel('t in sec. ');
25 ylabel('x(t)');
26 title('Continuous Time Signal');
27 figure
28 subplot(2,1,1);
29 a=gca();
30 a.y_location="origin";
31 plot(W,XW_Mag);
32 xlabel('Frequency in Radians/Seconds>W');
33 ylabel('abs(X(jW))');
34 title('Magnitude Response (CTFT)');
35 subplot(2,1,2);
36 a=gca();
37 a.y_location="origin";
38 a.x_location="origin";
39 plot(W,XW_Phase*%pi/180) ;
40 xlabel(' Frequency in Radians/ S e c o n d s           > W
      ');
41 ylabel('<X(jW)')
42 title('Phase Response (CTFT) in Radians');

```

---

## Chapter 4

# The Discrete Time Fourier Transform

Scilab code Exa 4.1 DTFT computation

```
1 //Example 4.1
2 //Find the DTFT of  $(a^n)u[n]$ , for  $|a|<1$ 
3 clc;
4 syms w a n;
5 x=a^n;
6 X=symsum(x*exp(-%i*w*n),n,0,%inf);
```

---

Scilab code Exa 4.2 DTFT computation

```
1 //Example 4.2
2 //Find DTFT of  $x[n]=(a^n)u[-(n+1)]$ 
3 clc;
4 syms w a n;
5 x=a^n;
6 X=symsum(x*exp(-%i*w*n),n,-%inf,-1);
```

---

**Scilab code Exa 4.3** DTFT computation of unit impluse

```
1 //Example 4.3
2 //Find DTFT of unit impluse
3 clc;
4 syms w n;
5 x=1;
6 X=symsum(x*exp(-%i*w*n),n,0,0);
```

---

**Scilab code Exa 4.5** DTFT computation

```
1 //Example 4.5
2 //Find DTFT of  $x[n]=a^{|n|}$  for  $-1<a<1$ 
3 clc;
4 syms w a n;
5 x1=a^n;
6 x2=a^(-n);
7 X1=symsum(x1*exp(-%i*w*n),n,0,%inf);
8 X2=symsum(x2*exp(-%i*w*n),n,-%inf,-1);
9 X=X1+X2;
10 disp(X, 'X(e^jw)=');
```

---

**Scilab code Exa 4.9** Sketch discrete time signal

```
1 //Example 4.9
2 clc;
3 syms w a n;
4 x=a^n;
5 pi=22/7;
```

```

6 X=symsum(x*exp(-%i*w*n),n,0,%inf);
7 n1=0:10;
8 a=0.5;
9 x1=a^n1;
10 plot2d3(n1,x1);
11 xtitle('Discrete Time Signal','n','x[n]');
12 a.thickness=2;

```

---

#### Scilab code Exa 4.10 DTFT of cosine

```

1 //Example 4.10
2 //Find DTFT of  $x[n]=\cos(\omega_0 n)$  with  $\omega_0=(2\pi/5)$ 
3 clc;
4 syms w n;
5 x1=exp(%i*(2*pi*n/5));
6 x2=exp(-%i*(2*pi*n/5));
7 X1=symsum(x1*exp(-%i*w*n),n,0,%inf);
8 X2=symsum(x2*exp(-%i*w*n),n,0,%inf);
9 X=(X1+X2)/2;

```

---

#### Scilab code Exa 4.12 DTFT of unit step

```

1 //Example 4.12
2 //Find DTFT of  $x[n]=u[n]$ 
3 clc;
4 syms w n;
5 x=1;
6 X=symsum(x*exp(-%i*w*n),n,0,%inf);

```

---

#### Scilab code Exa 4.16 DTFT computation

```
1 //Example 4.16
2 //Find DTFT of  $x[n]=(a^n)u[n]$ , for  $0<a<1$ 
3 clc;
4 syms w a n;
5 x=a^n;
6 X=symsum(x*exp(-%i*w*n),n,0,%inf);
```

---

**Scilab code Exa 4.22** DTFT computation

```
1 //Example 4.22
2 //Find DTFT of  $x[n]=((1/2)^{(n-1)})u[n-1]$ 
3 clc;
4 syms w n;
5 x=(1/2)^(n-1);
6 X=symsum(x*exp(-%i*w*n),n,1,%inf);
```

---



# Chapter 5

## Time and Frequency characterisation of signals and systems

Scilab code Exa 5.1 Bode Plot

```
1 //Example 5.1
2 //Obtain the Bode plot
3 clc;
4 s=%s;
5 H=syslin('c',2*10^4/(s^2+100*s+10^4));
6 bode(H,0.1,10000);
7 funcprot(0);
```

---

Scilab code Exa 5.2 Bode Plot

```
1 //Example 5.2
2 //Obtain the Bode plot
3 clc;
4 s=%s;
```

```
5 H=syslin('c',100*(1+s)/((10+s)*(100+s)));  
6 bode(H,0.01,2000);
```

---

# Chapter 6

## Sampling And Laplace Transform

Scilab code Exa 6.1 Find nyquist rate

```
1 //Example 6.1
2 clc;
3 disp('x(t)=3cos(50%pi*t)+10sin(300%pi*t)-cos(100%pi*
      t)');
4 w1=50*%pi;
5 w2=300*%pi;
6 w3=100*%pi;
7 f1=w1/(2*%pi);
8 f2=w2/(2*%pi);
9 f3=w3/(2*%pi);
10 if f1>f2 then
11     if f1>f3 then
12         disp(2*f1,'Nyquist rate=');
13     else
14         disp(2*f3,'Nyquist rate=');
15     end
16 else
17     if f2>f3 then
18         disp(2*f2,'Nyquist rate=');
```

```

19     else
20         disp(2*f3, 'Nyquist rate=');
21     end
22 end

```

---

#### Scilab code Exa 6.2 Find nyquist rate

```

1 //Example 6.2
2 clc;
3 disp('x(t)=(1/2*pi) cos(4000*pi*t) cos(1000*pi*t)');
4 w1=5000*pi;
5 w2=3000*pi;
6 f1=w1/(2*pi);
7 f2=w2/(2*pi);
8 if f1>f2 then
9     nyquist_rate=2*f1;
10 else
11     nyquist_rate=2*f2;
12 end
13 nyquist_interval=1/nyquist_rate;
14 disp(nyquist_rate, 'Nyquist rate=');
15 disp(nyquist_interval, 'Nyquist interval in seconds')
    ;

```

---

#### Scilab code Exa 6.4 Find nyquist rate

```

1 //Example 6.4
2 clc;
3 disp('x(t)=6 cos(50*pi*t)+20 sin(300*pi*t)-10 cos(100
    %pi*t)');
4 w1=50*pi;
5 w2=300*pi;
6 w3=100*pi;

```

```

7 f1=w1/(2*%pi);
8 f2=w2/(2*%pi);
9 f3=w3/(2*%pi);
10 if f1>f2 then
11     if f1>f3 then
12         disp(2*f1, 'Nyquist rate=');
13     else
14         disp(2*f3, 'Nyquist rate=');
15     end
16 else
17     if f2>f3 then
18         disp(2*f2, 'Nyquist rate=');
19     else
20         disp(2*f3, 'Nyquist rate=');
21     end
22 end

```

---

**Scilab code Exa 6.26** Laplace transform of signal

```

1 //Example 6.26
2 //Find laplace transform  $x(t)=2e^{-3t}u(t)-e^{-2t}u(t)$ 
3 clc;
4 syms t;
5 x=2*%e^(-3*t)-%e^(-2*t);
6 X=laplace(x);

```

---

**Scilab code Exa 6.27.a** Laplace transform of function

```

1 //Example 6.27a
2 //Laplace transform of  $x(t)=t^3+3*t^2-6*t+4$ 
3 clc;
4 syms t;

```

```
5 x=t^3+3*t^2-6*t+4;
6 X=laplace(x);
```

---

**Scilab code Exa 6.27.b** Laplace transform of function

```
1 //Example 6.27b
2 //x(t)=(cos(3t))^3
3 clc;
4 syms t;
5 x=(cos(3*t))^3;
6 X=laplace(x);
```

---

**Scilab code Exa 6.27.c** Laplace transform of function

```
1 //Example 6.27c
2 clc;
3 syms a b t;
4 x=sin(a*t)*cos(b*t);
5 X=laplace(x);
```

---

**Scilab code Exa 6.27.d** Laplace transform of function

```
1 //Example 6.27d
2 clc;
3 syms t a;
4 x=t*sin(a*t);
5 X=laplace(x);
```

---

**Scilab code Exa 6.27.e** Laplace transform of function

```
1 //Example 6.27e
2 clc;
3 syms t s;
4 x1=1-%e^t;
5 X1=laplace(x1);
6 X=integ(X1,s,s,%inf);
```

---

**Scilab code Exa 6.48** Find response of system

```
1 //Example 6.48
2 clc;
3 syms t s;
4 x=1+%e^(-3*t)-%e^(-t);
5 X=laplace(x);
6 H=1/((s+1)*(s^2+s+1));
7 Y=X*H;
8 y=ilaplace(Y);
```

---

# Chapter 7

## The Z Transform

Scilab code Exa 7.1 z transform

```
1 //Example 7.1
2 clc;
3 syms a z n;
4 x=a^n;
5 X=symsum(x*(z^-n),n,0,%inf);
6 disp(X, 'X(z)=');
```

---

Scilab code Exa 7.2 z transform of unit impulse

```
1 //Example 7.2
2 clc;
3 syms n z;
4 x=1;
5 X=symsum(x*(z^-n),n,0,0);
6 disp(X, 'X(z)=');
```

---



### Scilab code Exa 7.3 z transform of unit step

```
1 clc ;
2 syms n;
3 x=ones(1);
4 X=symsum(x*(z^-n),n,0,%inf);
5 disp(X, 'X(z)=');
```

---

### Scilab code Exa 7.5 z transform of cosine

```
1 //Example 7.5
2 clc ;
3 syms Wo n z;
4 x1=exp(sqrt(-1)*Wo*n);
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=exp(-sqrt(-1)*Wo*n);
7 X2=symsum(x2*(z^-n),n,0,%inf);
8 X=(X1+X2)/2;
9 disp(X, 'X(z)=');
```

---

### Scilab code Exa 7.6 z transform

```
1 //Example 7.6
2 clc ;
3 syms n z;
4 x=1;
5 X=symsum(x*(z^-n),n,-%inf,0);
6 disp(X, 'X(z)=');
```

---

### Scilab code Exa 7.7 z transform of sequence

```

1 //Example 7.7
2 clc;
3 syms n z;
4 X1=0;
5 X2=0;
6 for i=0:2:4
7     x1=(1/2)^i;
8     X1=X1+x1*z^-i;
9 end
10 for i=1:2:5
11     x2=(1/3)^i;
12     X2=X2+x2*z^-i;
13 end
14 x3=2^n;
15 X3=symsum(x3*(z^-n),n,-%inf,1);
16 X=X1+X2+X3;
17 disp(X, 'X(z)=');

```

---

#### Scilab code Exa 7.10 z transform

```

1 //Example 7.10
2 //Z-transform of (2^n)u[n-2]
3 clc;
4 syms n z;
5 x=2^n;
6 X=symsum(x*(z^-n),n,2,%inf);
7 disp(X, 'X(z)=');

```

---

#### Scilab code Exa 7.11.a z transform

```

1 //Example 7.11a
2 //Z transform of (a^n)cos(Wo*n)
3 clc;

```

```

4 syms Wo n z a;
5 x1=(a^n)*exp(sqrt(-1)*Wo*n);
6 X1=symsum(x1*(z^-n),n,0,%inf);
7 x2=(a^n)*exp(-sqrt(-1)*Wo*n);
8 X2=symsum(x2*(z^-n),n,0,%inf);
9 X=(X1+X2)/2;
10 disp(X, 'X(z)=');

```

---

#### Scilab code Exa 7.11.b z transform

```

1 //Example 7.11b
2 //Z transform of (a^n) sin(Wo*n)
3 clc;
4 syms Wo n z a;
5 x1=(a^n)*exp(sqrt(-1)*Wo*n);
6 X1=symsum(x1*(z^-n),n,0,%inf);
7 x2=(a^n)*exp(-sqrt(-1)*Wo*n);
8 X2=symsum(x2*(z^-n),n,0,%inf);
9 X=(X1-X2)/(2*%i);
10 disp(X, 'X(z)=');

```

---

#### Scilab code Exa 7.12 z transform using differentiation property

```

1 //Example 7.12
2 //Ztransform of x[n]=(n^2)*u[n] done by
   Differentiation property
3 clc;
4 syms z n;
5 x=1;
6 X1=symsum(x*(z^-n),n,0,%inf);
7 X2=(-z)*(diff(X1,z));
8 X=(-z)*(diff(X2,z));
9 disp(X, 'X(z)=');

```

---

**Scilab code Exa 7.13** z transform

```
1 //Example 7.13
2 //Differentiation property is used here
3 clc;
4 syms n z;
5 x1=((-1/2)^n);
6 x2=(1/4)^-n;
7 X1=symsum(x1*(z^-n),n,0,%inf);
8 X3=(-z)*diff(X1,z);
9 X2=symsum(x2*(z^-n),n,-%inf,0);
10 X=X3*X2;
11 disp(X,'X(z)=');
```

---

**Scilab code Exa 7.18** Find Discrete time input signal

```
1 //Example 7.18
2 //Determine the input x[n] if h[n]=[1,2,3] and y[n
   ]=[1,1,2,-1,3]
3 clc;
4 clear;
5 function [za]=ztransfer(sequence,n)
6     z=poly(0,'z','r')
7     za=sequence*(1/z)^n'
8 endfunction
9 z=poly(0,'z');
10 h=[1,2,3];
11 n1=0:length(h)-1;
12 H=ztransfer(h,n1);
13 y=[1,1,2,-1,3];
14 n2=0:length(y)-1;
```

```

15 Y=ztransfer(y,n2);
16 X=Y/H;
17 funcprot(0);
18 funcprot(0);
19 x=ldiv(z^2-z+1,z^2,3);
20 disp(x,'x[n]=');

```

---

#### Scilab code Exa 7.19 Inverse Z transform

```

1 //Example 7.19
2 //Find the inverse Z-transform using long division
  method
3 clc;
4 clear;
5 z=poly(0,'z');
6 x=ldiv(z^2,z^2-(3/2)*z+(1/2),4);
7 disp(x,'x[n]=');

```

---

#### Scilab code Exa 7.24.a Inverse Z transform using long division method

```

1 //Example 7.24a
2 //Inverse Z-transform using long division method
3 clc;
4 clear;
5 z=poly(0,'z');
6 x=ldiv(2*z^3+3*z^2,(z+1)*(z+0.5)*(z-0.25),4);
7 disp(x,'x[n]=');

```

---

#### Scilab code Exa 7.36 z transform

```

1 //Example 7.36
2 clc;
3 syms z n;
4 x1=2^n;
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=3^n;
7 X2=symsum(x2*(z^-n),n,0,%inf);
8 X=3*X1-4*X2;

```

---

**Scilab code Exa 7.37** z transform

```

1 //Example 7.37
2 clc;
3 syms z n;
4 x=(1/2)^n;
5 X=symsum(x*(z^-n),n,0,%inf);

```

---

**Scilab code Exa 7.38** z transform

```

1 //Example 7.38
2 clc;
3 syms a z n;
4 x=-(a^n);
5 X=symsum(x*(z^-n),n,-%inf,-1);

```

---

**Scilab code Exa 7.39** z transform using differentiation property

```

1 //Example 7.39
2 clc;
3 syms z n a;

```

```
4 x1=(a^n);
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 X=(-z)*(diff(X1,z));
```

---

Scilab code Exa 7.42.a z transform of sequence

```
1 //Example 7.42a
2 clc;
3 function [za]=ztransfer(sequence,n)
4     z=poly(0,'z','r')
5     za=sequence*(1/z)^n
6 endfunction
7 x=[1,2,3,4,5,0,7];
8 n1=0:length(x)-1;
9 X=ztransfer(x,n1);
10 funcprot(0);
```

---

Scilab code Exa 7.42.b z transform of sequence

```
1 //Example 7.42b
2 clc;
3 function [za]=ztransfer(sequence,n)
4     z=poly(0,'z','r')
5     za=sequence*(1/z)^n
6 endfunction
7 x=[1,2,3,4,5,0,7];
8 n1=-3:length(x)-4;
9 X=ztransfer(x,n1);
10 funcprot(0);
```

---

**Scilab code Exa 7.43** z transform

```
1 //Example 7.43
2 clc;
3 syms z n;
4 x1=(-1/3)^n;
5 x2=(1/2)^n;
6 X1=symsum(x1*(z^-n),n,0,%inf);
7 X2=symsum(x2*(z^-n),n,-%inf,-1);
8 X=X1-X2;
```

---

**Scilab code Exa 7.48.a** z transform

```
1 //Example 7.48a
2 clc;
3 syms z n;
4 x1=2^n;
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=(1/2)^n;
7 X2=symsum(x2*(z^-n),n,0,%inf);
8 X=X1+(3*X2);
```

---

**Scilab code Exa 7.48.b** z transform

```
1 //Example 7.48b
2 clc;
3 syms z n;
4 x1=(-1/2)^n;
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=(3)^n;
7 X2=symsum(x2*(z^-n),n,-%inf,-1);
8 X=(3*X1)-(2*X2);
```

---



**Scilab code Exa 7.50** z transform of sequence

```
1 //Example 7.50
2 clc;
3 syms n z;
4 X1=0;
5 X2=0;
6 for i=0:2:4
7     x1=(1/2)^i;
8     X1=X1+x1*z^-i;
9 end
10 for i=1:2:5
11     x2=(1/3)^i;
12     X2=X2+x2*z^-i;
13 end
14 x3=(2)^n;
15 X3=symsum(x3*(z^-n),n,-%inf,1);
16 X=X1+X2+X3;
```

---

**Scilab code Exa 7.52** z transform of discrete signal

```
1 //Example 7.52
2 //Z transform of  $x[n]=(2^n)u[n-2]$ 
3 clc;
4 syms z n;
5 x=2^n;
6 X=symsum(x*(z^-n),n,2,%inf);
```

---

**Scilab code Exa 7.54** Inverse Z transform

```

1 //Example 7.54
2 clc;
3 clear;
4 z=poly(0, 'z');
5 X=[2;3*z^-1;4*z^-2];
6 n=0:2;
7 ZI=z^n';
8 x=numer(X.*ZI);
9 disp(x, 'x[n]=');

```

---

**Scilab code Exa 7.56** Find Discrete time input signal

```

1 //Example 7.56
2 //Determine the input x[n] if h[n]=[1,2,3] and y[n]
   ]=[1,1,2,-1,3]
3 clc;
4 clear;
5 function [za]=ztransfer(sequence,n)
6     z=poly(0, 'z', 'r')
7     za=sequence*(1/z)^n'
8 endfunction
9 z=poly(0, 'z');
10 h=[1,2,3];
11 n1=0:length(h)-1;
12 H=ztransfer(h,n1);
13 y=[1,1,2,-1,3];
14 n2=0:length(y)-1;
15 Y=ztransfer(y,n2);
16 X=Y/H;
17 funcprot(0);
18 funcprot(0);
19 x=ldiv(1-z+z^2,z^2,3);
20 disp(x, 'x[n]=');

```

---

**Scilab code Exa 7.59.a** z transform

```
1 //Example 7.59 a
2 //Z transform of  $x[n]=-(a^n)u[-n-1]$ 
3 clc;
4 syms a n z;
5  $x=-(a^n)$ ;
6  $X=\text{symsum}(x*(z^{-n}),n,-\%inf,-1)$ ;
```

---

**Scilab code Exa 7.59.b** z transform

```
1 //Example 7.59 b
2 //Z transform of  $x[n]=(a^{-n})u[-n-1]$ 
3 clc;
4 syms a n z;
5  $x=(a^{-n})$ ;
6  $X=\text{symsum}(x*(z^{-n}),n,-\%inf,-1)$ ;
```

---

**Scilab code Exa 7.61.a** z transform of discrete signal

```
1 //Example 7.61 a
2 clc;
3 syms z n;
4  $x1=(1/2)^n$ ;
5  $X1=\text{symsum}(x1*(z^{-n}),n,0,\%inf)$ ;
6  $x2=(1/3)^n$ ;
7  $X2=\text{symsum}(x2*(z^{-n}),n,0,\%inf)$ ;
8  $X=X1+X2$ ;
```

---

**Scilab code Exa 7.61.b** z transform of discrete signal

```
1 //Example 7.61b
2 clc;
3 syms z n;
4 x1=(1/3)^n;
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=(1/2)^n;
7 X2=symsum(x2*(z^-n),n,-%inf,-1);
8 X=X1+X2;
```

---

**Scilab code Exa 7.61.c** z transform of discrete signal

```
1 //Example 7.61c
2 clc;
3 syms z n;
4 x1=(1/2)^n;
5 X1=symsum(x1*(z^-n),n,0,%inf);
6 x2=(1/3)^n;
7 X2=symsum(x2*(z^-n),n,-%inf,-1);
8 X=X1+X2;
```

---

**Scilab code Exa 7.65** z transform

```
1 //Example 7.65
2 clc;
3 syms z n;
4 h1=(1/2)^n;
5 H1=symsum(h1*(z^-n),n,0,%inf);
```

```
6 h2=(-1/4)^n;  
7 H2=symsum(h2*(z^-n),n,0,%inf);  
8 H=(H1+H2)/2;
```

---

**Scilab code Exa 7.68.a** z transform of discrete signal

```
1 //Example 7.68 a  
2 //Z transform of x[n]=u[n]  
3 clc;  
4 syms n z;  
5 x=1;  
6 X=symsum(x*(z^-n),n,0,%inf);
```

---

**Scilab code Exa 7.68.b** z transform of discrete signal

```
1 //Example 7.68 b  
2 //Z transform of x[n]=-u[-n-1]  
3 clc;  
4 syms n z;  
5 x=-1;  
6 X=symsum(x*(z^-n),n,-%inf,-1);
```

---

## Chapter 8

# Discrete Fourier Transform and Fast Fourier Transform

Scilab code Exa 8.1 Convolution of two finite duration sequences

```
1 //Example 8.1
2 //Determine the convolution of the two finite
   duration sequence
3 clc;
4 x=[1,1,1];
5 n1=-1:1;
6 h=[1,1,1];
7 n2=-1:1;
8 y=convol(x,h);
9 n=-2:1:2;
10 disp(y, 'y[n]= ');
11 a = gca ();
12 a.y_location =" origin";
13 a.x_location =" origin";
14 plot2d3(n,round(y),5);
15 poly1=a.children(1).children (1);
16 poly1.thickness=2;
17 xtitle('Plot of sequence y[n] ', 'n', 'y[n] ');
18 funcprot(0);
```

---

**Scilab code Exa 8.2** Responce of an FIR filter

```
1 //Example 8.2
2 //Find the response of an FIR filter with impulse
   response h[n]=[1,2,4] //to the input sequence x[n
   ]=[1,2]
3 clc;
4 x=[1,2];
5 h=[1,2,4];
6 Y=convol(x,h);
7 disp(Y, 'y[n]=');
```

---

**Scilab code Exa 8.3** DFT and IDFT

```
1 //Example 8.3
2 //Compute DFT of  $x(n) = \{1,1,0,0\}$  and IDFT of  $y(n) = \{1,0,1,0\}$ 
3 clc;
4 x=[1,1,0,0];
5 Y=[1,0,1,0];
6 X=fft(x, -1);
7 y=fft(Y, 1);
```

---

**Scilab code Exa 8.4** DFT computation

```
1 //Example 8.4
2 //Compute DFT of the following sequence
3 clc;
4 x=[0.25,0.25,0.25];
```

```
5 X=fft(x,-1);
6 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.5 DFT computation

```
1 //Example 8.5
2 //Find the DFT of the following sequence
3 clc;
4 x=[0.2,0.2,0.2];
5 n=-1:1;
6 X=fft(x,-1);
7 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.6 DFT of sequence

```
1 //Example 8.6
2 //Determine the DFT of the following sequence
3 clc;
4 x=[1,1,2,2,3,3];
5 X=fft(x,-1);
6 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.7 DFT computation

```
1 //Example 8.7
2 //DFT of  $x[n]=a.^n$ 
3 clc;
4 a=0.5; //Say for a=0.5
5 n=0:4;
6 x=a.^n;
```



```
7 X=fft(x,-1);
8 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.8 DFT computation

```
1 //Example 8.8
2 //Compute 4-point DFT of the sequence  $x[n]=\cos(n\pi/4)$ 
3 clc;
4 n=0:3;
5 pi=22/7;
6 x=cos(n*pi/4);
7 X=fft(x,-1);
8 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.9 IDFT computation

```
1 //Example 8.9
2 //Computing IDFT of the following sequence
3 clc;
4 X=[1,2,3,4];
5 x=fft(X,1);
6 disp(x,'x[n]=');
```

---

#### Scilab code Exa 8.10 IDFT computation

```
1 //Example 8.10
2 //Find the IDFT of the following sequence
3 clc;
4 i=sqrt(-1);
```

```
5 X=[3,2+i,1,2-i];
6 x=fft(X,1);
7 disp(x, 'x[n]=');
```

---

**Scilab code Exa 8.11** DFT computation using FFT algorithm

```
1 //Example 8.11
2 //For the given x[n] determine X[k] using FFT
  algorithm
3 clc;
4 x=[1,2,3,4,4,3,2,1];
5 X=fft(x,-1);
6 disp(X, 'X[k]=');
```

---

**Scilab code Exa 8.12** DFT computation using FFT algorithm

```
1 //Example 8.12
2 //For the given x[n] determine X[k] using FFT
  algorithm
3 clc;
4 x=[0,1,2,3,4,5,6,7];
5 X=fft(x,-1);
6 disp(X, 'X[k]=');
```

---

**Scilab code Exa 8.13** DFT computation

```
1 //Example 8.13
2 //Find the DFT of the following sequence
3 clc;
4 h=[1/3,1/3,1/3];
```

```
5 H=fft(h,-1);
6 disp(H,'H[k]=');
```

---

#### Scilab code Exa 8.14 DFT computation

```
1 //Example 8.14
2 //Find the DFT of the following sequence
3 clc;
4 h=[1/3,1/3,1/3];
5 H=fft(h,-1);
6 disp(H,'H[k]=');
```

---

#### Scilab code Exa 8.15 DFT computation

```
1 //Example 8.15
2 //Obtain the DFT of  $x[n]=(a^n) \cdot u[n]$ 
3 clc;
4 a=0.5;
5 for i=0:1:7
6     x(i+1)=a.^i;
7 end
8 X=fft(x,-1);
9 disp(X,'X[k]=');
```

---

#### Scilab code Exa 8.16 DFT computation

```
1 //Example 8.16
2 //Program to find DFT of sequence x
   =[1,1,1,1,1,1,1,0]
3 clc ;
```

```
4 x=[1,1,1,1,1,1,1,0];
5 X=fft(x,-1);
6 disp(X, 'X[k]= ');
```

---

Scilab code Exa 8.17 DFT computation

```
1 //Example 8.17
2 //Determine the DFT of the following sequence
3 clc;
4 x=[0,0,1,1,1,1,1,0,0,0];
5 X=fft(x,-1);
6 disp(X, 'X[k]= ');
```

---